Fermilab FERMILAB-Conf-01/227-E E918 August 2001

The BTeV DAQ and Trigger System - Some Throughput, Usability and Fault Tolerance Aspects

E.E. Gottschalk [1], J. Appel [1], T. Bapty [7], J. Barnes [7], T. Brennan [1], J. Butler [1], G. Cancelo[1], H. Cheung [1], D.C. Christian [1], M. Haney [3], R.K. Iyer [3], Z. Kalbarczyk [3], D.M. Kaplan [2], G. Karsai [7], P. Kasper [1], P. Lebrun [1], X.N. Li [2], P. McBride [1], D. Mosse [5], S. Neema [7], J. Oh [6], V. Pavlicek [1], R. Pordes [1], D. Petravick [1], M. Selen [3], P. Sheldon [7], S. Stone, W. Selove [4], M. Votava [1], M.H.L.S. Wang [1]

[1] *Fermi National Accelerator Laboratory, Batavia, IL 60510, USA*
[2] *Illinois Institute of Technology, Chicago, IL 60616, USA*
[3] *University of Illinois, Urbana-Champaign, IL 61801, USA*
[4] *University of Pennsylvania, Philadelphia, PA 19104, USA*
[5] *University of Pittsburgh, Pittsburgh, PA 15260, USA*
[6] *Syracuse University, Syracuse, NY 13244, USA*
[7] *Vanderbilt University, Nashville, TN 37235, USA*

Abstract

As presented at the last CHEP conference, the BTeV triggering and data collection pose a significant challenge in construction and operation, generating 1.5 Terabytes/second of raw data from over 30 million detector channels. We report on facets of the DAQ and trigger farms. We report on the current design of the DAQ, especially its partitioning features to support commissioning of the detector. We are exploring collaborations with computer science groups experienced in fault tolerant and dynamic real-time and embedded systems to develop a system to provide the extreme flexibility and high availability required of the heterogeneous trigger farm (~ ten thousand DSPs and commodity processors). We describe directions in the following areas: system modeling and analysis using the Model Integrated Computing approach to assist in the creation of domain-specific modeling, analysis, and program synthesis environments for building complex, large-scale computer-based systems; System Configuration Management to include compileable design specifications for configurable hardware components, schedules, and communication maps; Runtime Environment and Hierarchical Fault Detection/Management – a system-wide infrastructure for rapidly detecting, isolating, filtering, and reporting faults which will be encapsulated in intelligent active entities (agents) to run on DSPs, L2/3 processors, and other supporting processors throughout the system.

Keywords: data acquisition, real time, trigger

1 Physics and Detector

The recently approved BTeV experiment[1] is expected to begin running in the new Tevatron C0 interaction region at Fermilab by the year 2006. The physics goals include studies of CP violation and mixing, rare decays, and high sensitivity searches for decays forbidden within the Standard Model. The main focus of BTeV is on precision studies of CP violation and mixing in *B* decays.

The BTeV vertex trigger[2] is the primary physics trigger for the experiment. The trigger analyzes data for every interaction that occurs in the BTeV spectrometer, and selects *B* events by detecting the presence of detached beauty or charm decay vertices. It finds these vertices in the first stage of the trigger, Level 1 (L1). This means that the L1 trigger must perform the pattern recognition[3] as well as track and vertex reconstruction for every interaction at a rate of 15 million interactions per second. The L1 trigger must reject 99% of the background while selecting *B* events with high efficiency.

BTeV has detectors for charged-particle tracking, particle identification, E&M calorimetry, and muon detection. The charged-particle tracking consists of *vertex* and *forward* tracking systems. The vertex tracking system is a silicon pixel detector with 30 million pixels that provides data for the L1 vertex trigger. At Level 2 (L2) the vertex trigger uses data from the forward tracking system to

improve the tracks found at L1. At Level 3 (L3) the vertex trigger performs a complete analysis of data from the tracking systems. At this stage, the vertex trigger is one of several L3 trigger algorithms that process data to perform a physics-based analysis to identify interesting interactions.

2 DAQ and Trigger System

BTeV will generate an enormous amount of data at a rate of about 1.5 Terabytes per second. The factors that contribute to the large data rate are the interaction rate (15 million interactions/second), the large number of particles produced in the interactions, and the large number of detector channels (more than 30 million channels in the current design). Recording all of the data on archival media for later analysis is simply impossible. The solution is to analyze data from the spectrometer in real-time to select interesting *B* events, and to write these events to permanent storage for subsequent offline analysis. This challenging task is the purview of the BTeV DAQ and trigger system.

An overview of the architecture of the DAQ and trigger system is shown in Figure 1, and some significant numbers that characterize the system are given in Table 1. Figure 1 shows the buffers that receive data from BTeV detectors, an expanded view of the L1 vertex trigger, a Global L1 trigger that processes results from all first-level triggers, a switch that routes data to the Level 2/3 (L2/3) trigger, as well as the buffers and processors that make up the L2/3 trigger. No distinction is made between the hardware that is used for L2 and L3, since the same processors execute both the L2 and L3 algorithms. When a processor receives data for an interaction, the data are processed using the L2 algorithm. If the data satisfy L2 selection requirements, the data are processed using the L3 algorithm. Data that fail L2 or L3 selection requirements are dropped.

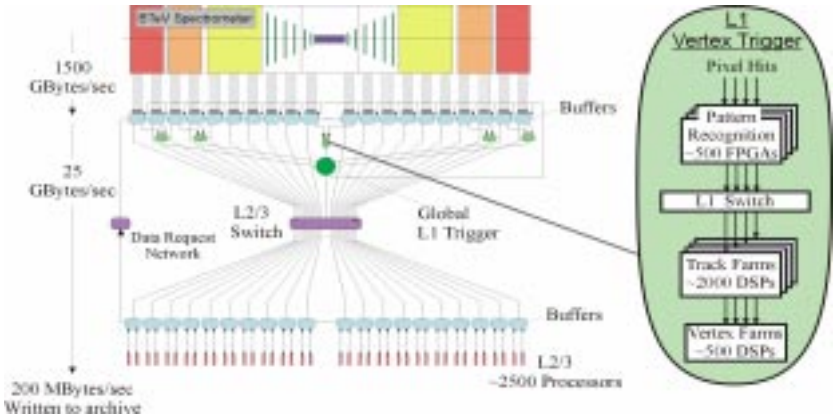

Figure 1: Schematic of the BTeV DAQ and trigger system showing (left side) the spectrometer, buffers, L2 and L3 processors and their interconnections and (right side) a blowup of the L1 vertex trigger.

Table 1: Characteristics of the BTeV DAQ and trigger system

# of Gbyte/s data links	Buffer memory	Data rate to L1 buffers	Data rate to L2/3 buffers	Data rate to archive	# of L1 DSPs	# of L2/3 processors
2500	1 Tbyte	1.5 Tbytes/s	25 Gbytes/s	200 Mbytes/s	> 2500	~ 2500

To estimate the scale of the DAQ and trigger project, we have performed a variety of Monte Carlo simulations, data-flow simulations and benchmarks for trigger algorithms. For the L1 vertex trigger we require about 500 Field Programmable Gate Arrays (FPGAs) to find track segments, and about 2500 Digital Signal Processors (DSPs) to reconstruct tracks and vertices. This estimate does not include additional processors for fault tolerance or other L1 algorithms, nor does it include the support processors required to configure, monitor, and control the DSPs. For the L2/3 trigger we require about 2500 general-purpose computers, such as Intel Pentiums running the LINUX operating system.

Requirements for the project are well understood, and the design is at an advanced conceptual stage. However, there is still enough flexibility in the design to adapt to new discoveries and different implementation strategies. For example, calculations that are done in DSPs may be migrated into FPGAs, or we may use more DSPs and fewer PCs, or vice versa. We anticipate that there will be variations in the design due to the availability of new types of hardware, the addition of redundant hardware for reliability, elimination of superfluous hardware, or new algorithms that require different hardware. The supporting software infrastructure must be flexible enough to handle design variations.

3 Usability and Fault Tolerance

While the hardware for the DAQ and trigger is extensive and complex and the trigger algorithms are challenging to develop, a greater challenge is to build a system that functions and produces quality results over a period of several years. The system must serve well during detector commissioning and debugging, during normal operation, troubleshooting, and calibration. Even during normal operation it must support several modes of operation and switch between modes dynamically. It must operate in spite of component failures. It must adapt to variations in the Tevatron accelerator and BTeV detector. It must evolve as hardware is replaced or as new components are introduced to extend its capabilities, and it must accommodate changes in software as more is learned about the physics. In addition to computational performance, the BTeV DAQ and trigger must support dynamic reconfiguration and partitioning, tolerate and adapt to fault conditions, and support maintainability and evolvability.

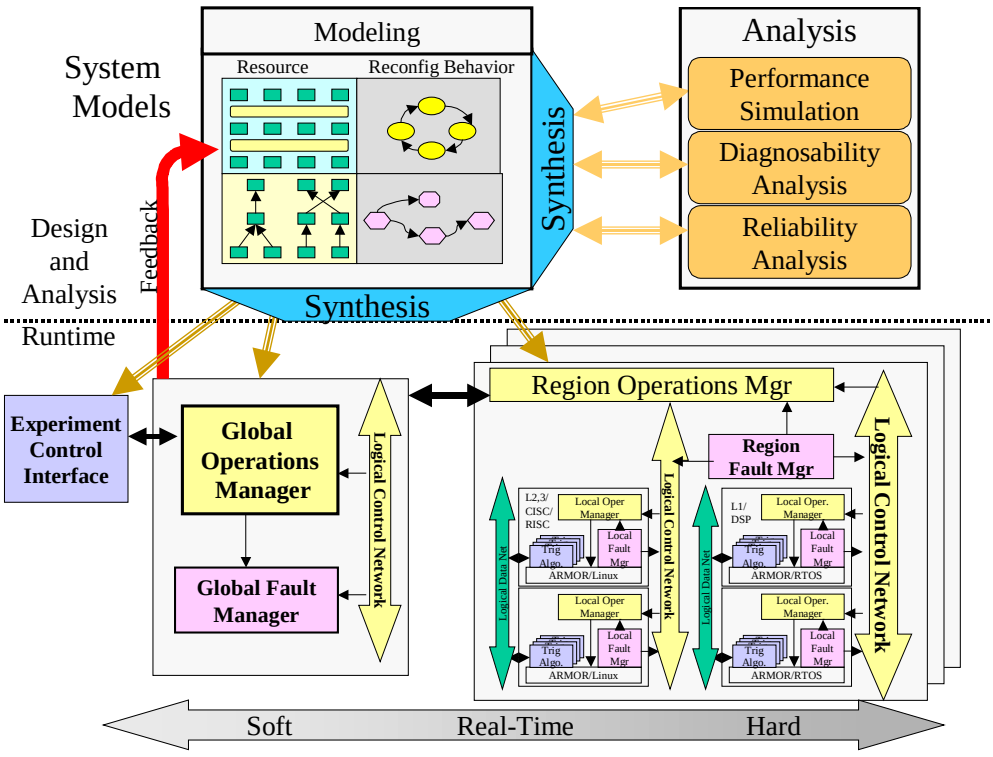

Figure 2: Bi-level system design and runtime framework

Dynamic reconfiguration and partitioning of the DAQ and trigger are important aspects of the design. To execute a number of different tasks (some of them simultaneously) the system must

operate in different modes. Example modes include: 1) normal trigger operation; 2) special modes for commissioning, debugging, and (re)calibration of the detector; 3) verification of repairs, and upgrades of hardware or software; 4) use of the system to test new algorithms; 5) verification of detector alignment and calibration at the start of each run; and 6) introduction of special diagnostic packages to investigate problems in the detector, the trigger hardware, or software.

System reliability and availability are exceedingly important. The system must be able to tolerate faults in network switching or processing elements. Under abnormal conditions, such as malfunctions that impair data quality or throughput, the system must continue to operate (if necessary, at decreased capacity). To achieve these goals, the system must autonomously detect fault conditions and respond in a manner that mitigates and adapts to faults. Due to the system's complexity, it could take a long time for an operator to recognize the existence of a problem, diagnose and remedy it. During this time valuable data could be lost, or be of poor quality. Operator intervention should be kept to a minimum.

We are exploring collaboration with computer science groups and are considering a variety of high-level design tools to develop the BTeV DAQ and trigger. Figure 2 illustrates a possible scenario in which system models use domain-specific, multi-view representations to define the behavior, performance, fault interactions, and hardware implementation of a system. Analysis tools are used to evaluate performance to guide hardware and software designers prior to system implementation, and synthesis tools are used to generate system configurations directly from the models. A fault-detecting and failure-mitigating runtime environment executes these configurations in the real-time, distributed, and heterogeneous DAQ and trigger hardware, with model-configured fault mitigation built in. Figure 2 depicts a scenario that entails local, regional, and global perspectives of the system. Moreover, the cooperation between runtime and modeling/synthesis environments permits global reconfiguration of hardware in response to failure conditions. Some of the concepts that we are investigating with regard to BTeV are Model Integrated Computing (MIC)[4], ARMORs [5] (Adaptive, Reconfigurable, and Mobile Objects for Reliability), and the use of the FT-RT-Mach [6] operating system.

4 Conclusion

The design and implementation of a high-performance real-time computational system that is reliable, flexible, fault-tolerant, and fault adaptive is certainly challenging. Although our efforts are directed towards the development of such a system for BTeV, the results should have applicability to other large parallel computational systems with very high reliability and availability requirements.

References

[1] Kulyavtsev, A., et al., "Proposal for an Experiment to Measure Mixing, CP Violation and Rare Decays in Charm and Beauty Particle Decays at the Fermilab Collider," (2000), http://www-btev.fnal.gov/public_documents/btev_proposal/.

[2] Gottschalk, E.E., et al., "BTeV Detached Vertex Trigger," FERMILAB-Conf-01-088-E, June 2001; and to be published in *Proceedings of the 9th International Workshop on Vertex Detectors (Vertex 2000)*, Homestead, MI, September 2000.

[3] An animation of the Level 1 pattern recognition with explanatory text can be found on the web: http://www-ppd.fnal.gov/btev_trigger/presentations/Animated_Trigger/.

[4] Franke, H., Sztipanovits, J., and Karsai, G., "Model-Integrated Computing," *Proceedings of the 1997 Hawaii Systems Sciences Conference*, (CD-ROM publication), 1997.

[5] Kalbarczyk, Z., Iyer, R.K., Bagchi, S., and Whisnant, K., "Chameleon: A Software Infrastructure for Adaptive Fault Tolerance," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 10, No. 6, pp. 560-579, June 1999.

[6] Egan, A., Kutz, D., Mikulin, D., Melhem, R., and Mosse, D., "Fault-Tolerant RT-Mach (FT-RT-Mach) and an Application to Real-Time Train Control," *Software Practice and Experience* (April 1999), Vol. 29, No. 4, pp. 379-395, Wiley Publishers.